



# A hybrid adaptive-gridding immersed-boundary lattice Boltzmann method for viscous flow simulations

Xixiong Guo<sup>a</sup>, Jieke Yao<sup>b</sup>, Chengwen Zhong<sup>c</sup>, Jun Cao<sup>a,\*</sup>

<sup>a</sup>Department of Mechanical and Industrial Engineering, Ryerson University, Toronto, Ontario M5B 2K3, Canada

<sup>b</sup>Chengdu Aircraft Industrial (Group) Co. Ltd., Chengdu, Sichuan 610092, PR China

<sup>c</sup>National Key Laboratory of Science and Technology on Aerodynamic Design and Research, Northwestern Polytechnical University, Xi'an 710072, PR China

## ARTICLE INFO

### Keywords:

Lattice Boltzmann method  
Immersed boundary method  
Feedback forcing model  
Adaptive mesh refinement  
“Bubble” interpolation function

## ABSTRACT

Robust use of adaptive mesh refinement (AMR) techniques in the immersed-boundary (IB) lattice Boltzmann method (LBM) framework is seldom reported, but indeed expected owing to its foreseeable broad applicability and computational efficiency. This study is aimed at developing a highly hybrid computational framework that seamlessly incorporates the AMR algorithm in the IB-LBM approach, so that challenging problems, including the case of an obstacle that moves through a flowing fluid, can be numerically investigated. Owing to the feedback forcing based IB model, the advantages, such as simple mechanics principle, explicit interpolations, and inherent satisfaction of no-slip boundary condition for solid surfaces, are fully exhibited. Additionally, the “bubble” function is employed in the local mesh refinement process so that, for newly generated nodes belonging to a region with overlapping coarse and fine cells, the solution of second-order accuracy can be obtained only through the spatial interpolation but no temporal interpolation. With simulation interests in both steady and unsteady flows around a single cylinder and bi-cylinders, a number of test cases performed in this study have demonstrated the usefulness and effectiveness of the present hybrid AMR-IB-LBM approach.

© 2015 Elsevier Inc. All rights reserved.

## 1. Introduction

Pioneered by Berger and Oliger [1], the adaptive mesh refinement (AMR) techniques, which feature dynamically optimizing the grid resolution over the entire computational domain in response to the numerical solution evolution, have proven to be an outperforming Computational Fluid Dynamics (CFD) simulation tool in terms of both reduction of computational expense and improvement of solution accuracy.

Historically, the AMR strategy first appeared in 1980s for solving hyperbolic conservation laws in two dimensions [1,2], which locally inserts finer rectangular subgrids into the underlying coarse mesh wherever a higher level of grid resolution is demanded. Inspired by the success of their AMR in its initial stage, the AMR techniques for fluid flow simulations were substantially developed in the class of conventional numerical methods, such as finite difference method (FDM) [3,4], finite volume method (FVM) [5,6], and finite element method (FEM) [7,8]. AMR methods mostly employ a uniform coarse mesh as initial or level-0 mesh, and then create new nodes to locally generate higher-level smaller-size cells in critical regions of the computational domain, which often naturally results in a set of hierarchical-tree-like data structure [9,10]. On the other

\* Corresponding author.

E-mail address: [jcao@ryerson.ca](mailto:jcao@ryerson.ca) (J. Cao).

hand, Ji et al. [11] recently proposed a cell-based structured adaptive mesh refinement (CSAMR) data structure to replace the tree structure with a hash table, which greatly helps reduce the computer memory required for the data structure storage use. In addition, the moving finite element (MFE) [12,13] and moving finite difference (MFD) methods [14,15] are also often considered belonging to the AMR approaches. The mesh moving methods allow grid nodes to migrate in response to the evolving solution while the total number of grid nodes keeps unchanged. Thus, the moving mesh methods are also viewed as a global node-redistribution approach.

In recent years, the lattice Boltzmann method (LBM) [16,17] has been studied as an alternative to the traditional Navier-Stokes solver. Compared to the conventional numerical methods, LBM is relatively new, but noticeably advantageous in terms of simplicity for implementation, parallelizability for algorithmic development, and robustness for applications [18–20]. In the early stage of LBM development, structured Cartesian meshes with uniform square cells were routinely employed. Obviously, uniform mesh is mostly an inappropriate option especially for intricate flow problems that evolve with rapidly varying fluid properties in sensitive zones of the computational domain; therefore, the applicability of non-uniform meshes to the LBM framework is demanded for more effective and economical computations.

In order to make LBM executable on a non-uniform mesh, a variety of techniques were sought. Apart from the unstructured mesh [21], multi-block mesh [22], and meshless [23] methods, the incorporation of the AMR strategy in the LBM framework has been intensively studied. Crouse et al. [24] employed a quadtree-based data structure for coupling AMR with LBM. Neumann and Neckel [25] developed an AMR-LBM model based on the octree data structure. Tolke et al. [26] applied the non-uniform tree-type adaptive mesh in LBM for multi-phase flow simulations. Yu and Fan [27] used the PARAMESH toolkit [28] for AMR-LBM simulation of the buoyancy-driven bubble rising phenomena. All these cited pieces of AMR-LBM work used the tree-type data structure to manage a hierarchy of adaptive grids at different resolution levels. Wu and Shu [29] employed an efficient stencil adaptive LBM method originally proposed by Ding and Shu [30] in the finite difference context. In that method, two types of 5-point symmetric stencils construct a 9-point configuration, which is similar to the D2Q9 lattice. More recently, Guzik et al. [31] proposed an interesting method for adaptively refining the LBM mesh, featuring the use of finite-volume techniques for transferring information across the interface between different grid resolution zones as well as the use of a space-time interpolation scheme for ensuring the mass and momentum conservation. However, most of the above-cited AMR-LBM schemes require the solutions at different time steps for the interpolation purpose. This implicit nature results in the request of storing additional data together with the consumption of an undesirably great amount of computation time, which adversely affects the LBM efficiency.

Relevant to a large number of industrial and daily life applications, the interaction between fluid and moving object has become an attractive research subject. In order to numerically investigate this class of interaction problems, the immersed boundary (IB) method was originally proposed by Peskin [32]. In 2002, the IB approach was incorporated in LBM the first time by Feng et al. [33,34], which introduces an additional “restoring” force in the vicinity of the immersed object for considering the effect of the solid boundary, and then this additional force will be distributed to the Cartesian grid nodes as a special local body force. However, some IB-LBM models [33,34] are unable to maintain the no-slip boundary condition for the immersed boundary. Also, most existing IB-LBM methods [35,36] rely on implicit schemes to account for the effect of the restoring force. The weakness of implicit scheme manifests mainly in excessive computational expenses. Nevertheless, IB-LBM deserves great recommendation since in principle this approach does not demand mesh re-adjustment no matter how the immersed body moves or deforms and, obviously, it would be even more beneficial if IB-LBM and AMR can be coupled together. However, the effective AMR-IB-LBM incorporation is rarely found in available literature.

In summary, in coupling AMR with IB-LBM, not only are interpolation techniques requested at new nodes generated at the interface between the coarse and fine grid zones in the AMR-LBM context, but also for allocation of the restoring force in the IB-LBM framework. Although the implicit schemes are known as good for improving the interpolation accuracy, they inevitably incur a considerable loss in terms of computational efficiency. Thus, the most challenging issue in the AMR-IB-LBM coupling is to seek interpolation methods that are both accurate and economical.

Fortunately, a new interpolation strategy was proposed by Geier et al. [37] for LBM mesh refinement, which relies on a “bubble” function that eliminates the need of time interpolation and, thus, results in an explicit interpolation scheme. Another advantage of the bubble interpolation method manifests in its second-order numerical accuracy. On the other hand, to circumvent the reliance of IB on implicit scheme, Huang and Sung [38] derived a novel IB model that was based upon the feedback forcing principle and embedded in a finite difference flavored Navier-Stokes solver. It turns out to be an explicit scheme for obtaining the restoring force term directly from the feedback law when simulating the interaction between the flowing fluid and an object immersed in the fluid. The resulting feedback law based IB scheme essentially precludes the flowing fluid from penetrating the boundary of the solid object, leading to inherent satisfaction of the no-slip boundary condition. Compared to other IB-LBM models, this fully explicit IB-LBM model tremendously facilitates the entire solution procedure.

Inspired by the reported success of the bubble interpolation and feedback forcing methods, these two numerical strategies are both adopted in this study. Of more importance is to re-tailor these two innovative models such that a hybrid numerical method with significantly enhanced capabilities, as the goal of this study, can be developed that enables the AMR algorithm to be successfully implemented within the IB-LBM conjunctive framework and aims at simulations of complex fluid flow problems.

The rest of this paper is arranged as follows: Section 2 explains the principles of the IB-LBM approach. Section 3 will describe the main aspects of the AMR algorithm developed by this study. A number of benchmark test cases will be

examined in Section 4, and the detailed results will demonstrate the validity of the present hybrid model. Finally, some concluding and outlook remarks drawn from this study can be found in Section 5.

## 2. Immersed boundary–Lattice Boltzmann method (IB–LBM)

### 2.1. Lattice-Boltzmann method (LBM)

Different from the Navier–Stokes solver, LBM is derived from the kinetic theory, consisting of three ingredients: a discrete velocity model defined at each node of a lattice, a set of equilibrium distribution functions, and a time evolution equation. The lattice Boltzmann equation (LBE) reads:

$$f_a(\vec{x} + \vec{e}_a \Delta t, t + \Delta t) - f_a(\vec{x}, t) = \Omega_a, \quad (1)$$

in which  $f_a$  is the particle distribution function,  $\Omega_a$  is the collision operator, and  $\vec{e}_a$  is the particle velocity in the  $a$  th direction. In the present study, the two-dimension nine-velocity (D2Q9) model [16] is applied for describing the discrete particle velocity, which is defined as:

$$\vec{e}_a = \begin{cases} \vec{0} & a = 0 \\ c(\cos[(a-1)\pi/2], \sin[(a-1)\pi/2])^T & a = 1, 2, 3, 4 \\ \sqrt{2}c(\cos[(a-9/2)\pi/2], \sin[(a-9/2)\pi/2])^T & a = 5, 6, 7, 8 \end{cases} \quad (2)$$

The lattice speed  $c = \delta x / \delta t$  takes 1, where  $\delta x$  is the lattice size ( $\delta x = \delta y$  in the case of two-dimensional square lattice) and  $\delta t$  is the time step. When using the traditional single relaxation time (SRT) model, the collision operator is expressed by:

$$\Omega_a = -\frac{1}{\tau} (f_a(\vec{x}, t) - f_a^{eq}(\vec{x}, t)), \quad (3)$$

where  $\tau$  is the single relaxation parameter, and  $f_a^{eq}(\vec{x}, t)$  is the equilibrium distribution function, which can be expressed as:

$$f_a^{eq}(\vec{x}, t) = \rho w_a \left[ 1 + \frac{\vec{e}_a \cdot \vec{u}}{c_s^2} + \frac{(\vec{e}_a \cdot \vec{u})^2 - c_s^2 |\vec{u}|^2}{2c_s^4} \right], \quad (4)$$

with the weighting factor  $w_a$  defined as:  $w_0 = \frac{4}{9}$ ,  $w_1 = w_3 = w_5 = w_7 = \frac{1}{9}$ , and  $w_2 = w_4 = w_6 = w_8 = \frac{1}{36}$ , and  $c_s = \frac{c}{\sqrt{3}}$  representing the sonic lattice speed.

Although the SRT model has been routinely used in many applications, this model may lead to numerical instability when the relaxation time is close to 0.5. In order to strengthen the LBM numerical stability as well as ensure the full applicability of mesh adaptive techniques in the LBM framework, the following multiple-relaxation-time (MRT) LBM model [17] is employed in the present study:

$$\mathbf{f}(\vec{x} + \vec{e}_a \Delta t, t + \Delta t) = \mathbf{f}(\vec{x}, t) - \mathbf{M}^{-1} \mathbf{S} [\mathbf{m}(\vec{x}, t) - \mathbf{m}^{eq}(\vec{x}, t)], \quad (5)$$

where the moment space  $\mathbf{m}$  is related to the distribution function by a  $9 \times 9$  transformation matrix  $\mathbf{M}$  as follows:

$$\mathbf{m} = \begin{pmatrix} \rho \\ e \\ \varepsilon \\ j_x \\ q_x \\ j_y \\ q_y \\ p_{xx} \\ p_{xy} \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ -4 & -1 & -1 & -1 & -1 & 2 & 2 & 2 & 2 \\ 4 & -2 & -2 & -2 & -2 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & -1 & 0 & 1 & -1 & -1 & 1 \\ 0 & -2 & 0 & 2 & 0 & 1 & -1 & -1 & 1 \\ 0 & 0 & 1 & 0 & -1 & 1 & 1 & -1 & -1 \\ 0 & 0 & -2 & 0 & 2 & 1 & 1 & -1 & -1 \\ 0 & 1 & -1 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 \end{pmatrix} \begin{pmatrix} f_0 \\ f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \\ f_7 \\ f_8 \end{pmatrix} = \mathbf{M} \mathbf{f}, \quad (6)$$

and  $\mathbf{m}^{eq}$  groups the corresponding equilibrium values as follows:

$$\mathbf{m}^{eq} = \rho \left( 1, -2 + 3u^2, 1 - 3u^2, u_x, -u_x, u_y, -u_y, u_x^2 - u_y^2, u_x u_y \right)^T. \quad (7)$$

The relaxation matrix  $\mathbf{S}$  is a diagonal one:

$$\mathbf{S} = \text{diag}(s_0, s_1, \dots, s_8) = \text{diag}(0, s_e, s_\varepsilon, 0, s_q, 0, s_v, s_v), \quad (8)$$

where  $s_0 = s_3 = s_5 = 0$  is permissible, and recommended for reducing the computational cost. The relaxation rates,  $s_e$ ,  $s_q$ ,  $s_\varepsilon$ , are adjustable within the interval of  $(0, 2)$ ,  $s_v$  and  $s_e$  are related to the kinematic viscosity  $\nu$  and bulk viscosity  $\zeta$ :

$$\nu = c_s^2 \left( \frac{1}{S_v} - \frac{1}{2} \right) \delta_t, \quad (9)$$

$$\zeta = c_s^2 \left( \frac{1}{S_e} - \frac{1}{2} \right) \delta_t. \quad (10)$$

From the conservation laws of mass and momentum, the macroscopic density,  $\rho$ , velocity,  $\vec{u}$ , and pressure,  $p$ , which are their respective nodal values in the LBM framework, are defined with the aid of distribution function as follows:

$$\rho = \sum_a f_a, \quad (11)$$

$$\rho \vec{u} = \sum_a \vec{e}_a f_a, \quad (12)$$

$$p = \rho c_s^2. \quad (13)$$

## 2.2. Immersed boundary (IB) method embedded in LBM

In the IB approach, the flow field is represented by a set of Eulerian points, which are in fact the fixed Cartesian mesh nodes in the LBM context, and the boundary of a curved object, which is immersed in the fluid field, is represented by a set of Lagrangian points. These two sets of points in the IB–LBM framework are illustrated in Fig. 1. The fundamental idea of IB is that the effect of the fluid to the object boundary is modeled by a “restoring” force that should be additionally placed in the governing equations.

The governing equations of the conventional IB scheme for an incompressible viscous fluid flow can be written as [34]:

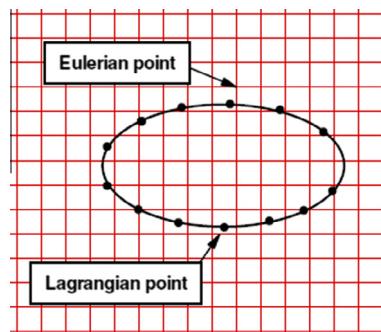
$$\rho \left( \frac{\partial \vec{u}}{\partial t} + (\vec{u} \cdot \vec{\nabla}) \vec{u} \right) + \vec{\nabla} p = \mu \Delta \vec{u} + \vec{f}_{BDR}, \quad (14)$$

$$\vec{\nabla} \cdot \vec{u} = 0, \quad (15)$$

$$\vec{f}_{BDR}(\vec{x}, t) = \int_{\Gamma} \vec{F}_{BDR}(s, t) \delta(\vec{x} - \vec{X}(s, t)) ds, \quad (16)$$

$$\frac{\partial \vec{X}(s, t)}{\partial t} = \vec{U}_B(\vec{X}(s, t), t) = \vec{u}(\vec{X}(s, t), t) = \int_{\Omega} \vec{u}(\vec{x}, t) \delta(\vec{x} - \vec{X}(s, t)) d\vec{x}, \quad (17)$$

where  $\vec{x}$  and  $\vec{u}$  are the Eulerian coordinates and fluid velocity;  $\vec{f}_{BDR}$  represents the force density acting on the fluid due to the immersed boundary, which is also referred to as “restoring force”;  $s$  represents the Lagrangian parametric coordinates;  $\rho$  and  $p$  stand for the fluid density and fluid pressure, respectively;  $\vec{X}$  and  $\vec{U}_B$  are the coordinates and velocity of the Lagrangian points;  $\vec{F}_{BDR}$  is the density of the boundary force strictly distributed along the immersed boundary, which is also referred to as “Lagrangian force”; and  $\delta(\vec{x} - \vec{X}(s, t))$  is the Dirac delta function. Eqs. (14) and (15) are the Navier–Stokes Equations with external force. To describe the interaction between the immersed object and the flowing fluid, Eq. (16) serves to distribute the boundary Lagrangian force, which acts at the Lagrangian points, to the Eulerian points, and Eq. (17) plays a role of interpolating the velocity, which takes place at the Eulerian points, to the Lagrangian points. It should be also remarked that, in Eq. (17), the no-slip boundary condition at the immersed solid boundary is fully guaranteed by the identity  $\vec{U}_B(\vec{X}(s, t), t) = \vec{u}(\vec{X}(s, t), t)$ .



**Fig. 1.** Schematic of IB–LBM concept.

In order to couple the IB approach with the LBM framework, Eqs. (14) and (15) can be adapted such that the hybrid IB–LBM model [39] can be generated as follows:

$$f_a(\vec{x} + \vec{e}_a \Delta t, t + \Delta t) - f_a(\vec{x}, t) = -\frac{1}{\tau} (f_a(\vec{x}, t) - f_a^{eq}(\vec{x}, t)) + (\tilde{f}_{BDR})_a \Delta t, \quad (18)$$

$$(\tilde{f}_{BDR})_a = \left(1 - \frac{1}{2\tau}\right) w_a \left( \frac{\vec{e}_a - \vec{u}}{c_s^2} + \frac{\vec{e}_a \cdot \vec{u}}{c_s^4} \vec{e}_a \right) \cdot \vec{f}_{BDR}, \quad (19)$$

$$\rho \vec{u} = \sum_a \vec{e}_a f_a + \frac{1}{2} \vec{f}_{BDR} \Delta t. \quad (20)$$

Note that, in the above hybrid IB–LBM model,  $(\tilde{f}_{BDR})_a$  represents one component of  $\vec{f}_{BDR}$ , which is transformed, via Eq. (19), from the restoring force,  $\vec{f}_{BDR}$ , to fit in with the LBM framework.

In an incompressible viscous fluid flow field  $\Omega$ , let  $\Gamma, \vec{x}, \vec{X}_B^l$ , ( $l = 1, 2, \dots, m$ ) denote the boundary of an immersed object, the Eulerian and Lagrangian points, respectively. According to Eq. (17), the discretized form of the velocity at a Lagrangian point denoted by the superscript  $l$  can be expressed as:

$$\vec{U}_B^l(\vec{X}_B^l, t) = \sum_{ij} \vec{u}(\vec{x}, t) D_{ij}(\vec{x}_{ij} - \vec{X}_B^l) \Delta x \Delta y, \quad (21)$$

in which  $D_{ij}(\vec{x}_{ij} - \vec{X}_B^l)$ , as replacement of the Dirac function  $\delta(\vec{x} - \vec{X}(s, t))$  in Eq. (17) for measuring the closeness of an Eulerian point,  $\vec{x}_{ij} = (x_i, y_j)^t$ , and a Lagrangian point,  $\vec{X}_B^l = (X_B^l, Y_B^l)^t$ , takes the following approximation:

$$D_{ij}(\vec{x}_{ij} - \vec{X}_B^l) = \delta(x_i - X_B^l) \delta(y_j - Y_B^l), \quad (22)$$

where  $\delta(r)$  is a continuous function defined as follows for all real  $r$ :

$$\delta(r) = \begin{cases} \frac{1}{4} \left(1 + \cos\left(\frac{\pi|r|}{2}\right)\right), & |r| \leq 2, \\ 0, & |r| > 2, \end{cases} \quad (23)$$

which was originally proposed by Peskin [32].

Note that Eq. (21) applies specifically to the case of a stationary body immersed in the flowing fluid. For the sake of more generality, in the case where the immersed body itself is moves at a velocity of  $\vec{U}^l$ , Eq. (21) should be revised to:

$$\vec{U}_B^l(\vec{X}_B^l, t) = \vec{U}^l + \sum_{ij} \vec{u}(\vec{x}, t) D_{ij}(\vec{x}_{ij} - \vec{X}_B^l) \Delta x \Delta y, \quad (24)$$

to ensure the satisfaction of the no-slip boundary condition, where  $\vec{U}^l$  represents the motion of the object.

The approximation of the Dirac function, as shown in Eq. (22), is also employed in determining the restoring force,  $\vec{f}_{BDR}$ , based on the Lagrangian force,  $\vec{F}_{BDR}$ , as follows:

$$\vec{f}_{BDR}(\vec{x}_{ij}, t) = \sum_l \vec{F}_{BDR}(\vec{X}_B^l, t) D_{ij}(\vec{x}_{ij} - \vec{X}_B^l) \Delta s_l, \quad (25)$$

where  $\Delta s_l$  is the arc length of a boundary element. In practice, the following feedback law based formula [38] is adopted in this study in order to obtain the force exerted at the Lagrangian point:

$$\vec{F}_n^l = \alpha \sum_{j=1}^n \left( (\vec{U}_B^l)_j - (\vec{U}^l)_j \right) \Delta t + \beta \left( (\vec{U}_B^l)_n - (\vec{U}^l)_n \right), \quad l = 1, 2, \dots, m. \quad (26)$$

In Eq. (26), the superscript  $l$  represents the Lagrangian point at the boundary of the immersed object, the subscript  $j$  denotes the  $j$ th time step,  $n$  stands for the current time point;  $\vec{U}_B^l$  is the velocity of the boundary point  $l$ , which has been calculated using Eq. (21) or (24);  $\vec{U}^l$  is the velocity of the obstacle immersed in the fluid, as previously described for Eq. (24), and its numerical value is calculated using  $(\vec{U}^l)_j = \frac{(\vec{X}^l)_j - (\vec{X}^l)_{j-1}}{\Delta t}$  (apparently, for a stationary object,  $\vec{U}^l \equiv \vec{0}$  since the entire object including its boundary does not move at all);  $\alpha$  and  $\beta$  are two negative free constants with dimensions of  $ML^{-3}T^{-2}$  and  $ML^{-3}T^{-1}$ , respectively. According to [30],  $\alpha$  and  $\beta$  are stable for moderate values within the interval  $[-100, -1]$ , and neither is sensitive to its exact value.

The IB–LBM solution procedure can be summarized as follows:

- (1) Set initial values for variables, including the initialization for the restoring force density at  $\vec{f}_{BDR} = \vec{0}$ .
- (2) Use Eq. (18) to get the distribution function at time  $t = t_n$  (initially  $\vec{f}_{BDR} = \vec{0}$ ), then compute the macroscopic variables, such as the density, pressure and velocity using Eqs. (11)–(13).
- (3) Obtain the Lagrangian force density using Eq. (21) or (24), and then Eq. (26).

- (4) Use Eq. (25) to determine the restoring force at the Eulerian points that are affected by the Lagrangian force due to the immersed boundary.
- (5) Compute the equilibrium distribution function using Eq. (4).
- (6) Repeat Steps (2) to Step (5) until a prescribed convergence criterion is satisfied.

### 3. Adaptive mesh refinement (AMR) techniques

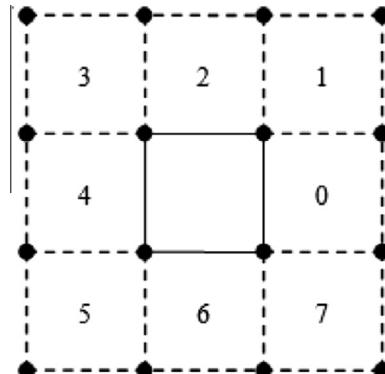
#### 3.1. Underlying mesh

The AMR algorithm developed in this study aims to generate an unstructured adaptive mesh in the LBM framework that pertinently responds to the progressive change of fluid flow pattern. The mesh adaptation is based upon a structured mesh, which is a uniform coarse Cartesian mesh and referred to as “underlying” mesh or “level-0” mesh. Two principal sets of data structures are required to describe the underlying mesh: one set for cells, and the other for nodes.

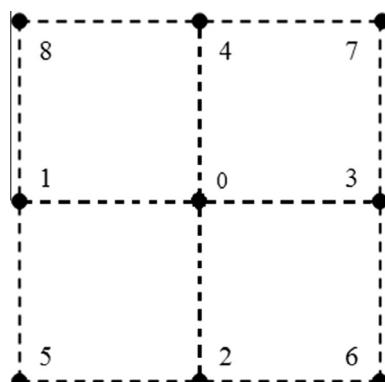
For defining a cell, the node numbers of its four vertices and its neighboring cell numbers, which are the basic information of the connectivity, has to be recorded. The convention used to locally number its eight neighboring cells is depicted in Fig. 2, in which the local neighboring element number increases following the counter-clockwise order. Meanwhile, a set of data structure for defining all nodes on the mesh should be also built, which includes, for each node, its coordinates, macroscopic variables such as the velocity, density, distribution function used in the LBM process, and the numbers of its eight neighboring nodes. Specific to a node, the local numbering convention for its eight surrounding nodes is illustrated in Fig. 3. In particular, when a boundary cell or node is processed, some of its neighboring nodes and cells do not exist on the underlying mesh and, accordingly, get an assigned value of 0 in the two sets of data structure.

#### 3.2. Mesh adaptive criterion

The adaptive gridding procedure may start after obtaining the numerical solution on the level-0 uniform mesh. The dynamic mesh refinement is controlled by a monitoring parameter, which is also referred to as “mesh adaptive criterion”. This



**Fig. 2.** Numbering of neighboring cells around an underlying cell.



**Fig. 3.** Numbering of neighboring nodes surrounding an underlying node.

criterion is expected to play a role of locating the zone of primary flow characteristics, which is also referred to as “critical region”. Hence, the selected criterion should be sufficiently sensitive to the important features of a particular flow problem. In reported adaptive LBM simulations of flow problems, different types of mesh adaptive criteria were proposed. Particularly, the divergence of the velocity [24] was more frequently employed for guiding mesh adaptation. Extended from that selection, this adaptive-gridding IB-LBM study relies on the first- or second-order derivatives of the velocity components to determine the AMR regions.

For the steady-state flow, the criterion,  $\varepsilon_1$ , which is defined as follows:

$$\varepsilon_1 = \sqrt{\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial u}{\partial y}\right)^2 + \left(\frac{\partial v}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial y}\right)^2} \quad (27)$$

is used to guide the mesh adaptation. The criterion  $\varepsilon_1$  takes the gradients of the velocity components  $u$  and  $v$  as ingredients. This criterion is calculated on a node-by-node basis using the velocity solution over the neighborhood of the node. With  $\bar{\varepsilon}_1$  denoting the average of  $\varepsilon_1$  over the entire computational domain, wherever  $\varepsilon_1 > \bar{\varepsilon}_1$  takes place at a node, its neighboring cells will be locally refined.

For steady flows,  $\varepsilon_1$  is considered pertinent, because this velocity gradient based criterion is capable of capturing the zones where the flow rate changes most significantly, which usually represents the most remarkable characteristics of steady flows. For instance, in the flow past a cylinder at  $Re = 40$ , which will be simulated in the present study (see Section 4.2), it is well-known that a pair of symmetric eddies take place behind the cylinder, and these re-circulation zones corresponding to elevated velocity gradients can be unquestionably detected by  $\varepsilon_1$ . However, in the case of an unsteady flow, for example, when the flow past a cylinder reaches  $Re = 100$ , which will be examined in Section 4.3, the famous Karman vortex street arises and evolves in an unsteady fashion. Ideally, the mesh adaptive criterion is expected to catch up with the vortex-shedding periodicity, but  $\varepsilon_1$  looks incompetent in this case since it focuses on the change of velocity only. By noticing that the vorticity variation, in lieu of the change of flow rate, becomes the dominant factor for unsteady flows, the gradient of the vorticity, which is essentially the second-order derivative of the velocity, looks more pertinent in detecting the solution sensitivity than  $\varepsilon_1$  can do. Therefore, another mesh adaptive criterion,  $\varepsilon_2$ , which takes the following form:

$$\varepsilon_2 = \sqrt{\left(\frac{\partial^2 u}{\partial x^2}\right)^2 + \left(\frac{\partial^2 u}{\partial y^2}\right)^2 + \left(\frac{\partial^2 v}{\partial x^2}\right)^2 + \left(\frac{\partial^2 v}{\partial y^2}\right)^2} \quad (28)$$

is suggested for the vortex shedding cases. Generally, in the present AMR method, given the two options  $\varepsilon_1$  and  $\varepsilon_2$ , the choice of mesh adaptive criterion depends upon the time-dependency nature of a particular flow.

### 3.3. Adaptive mesh refinement (AMR) algorithm

After the LBM solution is obtained on the level-0 mesh, the computation of the defined monitoring parameter, i.e., either  $\varepsilon_1$  or  $\varepsilon_2$ , is looped over all nodes on the mesh, and the average value of this parameter can be obtained, which becomes the threshold for local LBM cell refinement. For a cell that has the monitoring parameter exceeding the threshold, the “central refinement” rule should be applied. That is, a new square cell with an area of one-quarter, which is named “level-1” cell, will be placed at the central of the original cell that has a unit area as the edges of a level-0 cell are  $\delta x = \delta y = 1$ . Thus, if the local coordinates of the center of a level-0 cell take  $(0.5, 0.5)$ , the four new nodes of the level-1 cell will be located at  $(0.25, 0.25)$ ,  $(0.25, 0.75)$ ,  $(0.75, 0.25)$  and  $(0.75, 0.75)$ , respectively, as shown in Fig. 4. This illustrates a typical “parent and central-child” relation, which generates a new “refined cell tree” data structure accompanying the AMR implementation. Such a tree imparts logical structure to the mesh, and also potentially allows for simple un-refinement (tree-pruning) algorithms.

The central refinement is the direct response to the case where the monitoring parameter of a cell exceeds the mean monitoring parameter value, which essentially reflects the presence of solution sensitivity. After all necessary central refinements are implemented, the “inserting refinement” rule should be applied in the present AMR algorithm to generate, if necessary, supplementary refined cells between the central refined cells. That is, when two horizontally neighboring level-0 cells are both refined due to their respective unacceptably large monitoring parameter values, an additional level-1 cell must be inserted to bridge the two central level-1 cells, as illustrated in Fig. 5. Likewise, Fig. 6 shows the application of the inserting refinement rule to the vertical case.

As a result of the sequential use of the central and, then, inserting refinement rules, typical refinement patterns are illustrated in Figs. 7 and 8, which are viewed restrictively in the region composed of four level-0 cells that are neighbored with each other and share a common vertex. In Fig. 7, the four underlying cells are all selected for central refinement, followed by two horizontal and two vertical inserting refinements. Hence, in addition to the eight level-1 cells that have been generated by applying the central and inserting refinement rules, an adjoint level-1 cell in the central of this region is naturally created. This is referred to as “vortex refinement” rule, which is named after the fact that a level-0 vortex is located at the center of the adjoint level-1 cell. However, in some cases, the four level-0 cells are not necessarily all chosen for central refinement. Fig. 8 shows a case where the central refinement is implemented only in three neighboring underlying cells such that only one horizontal and one vertical inserting refinements are required, resulting in an “L-shaped” level-1 refinement pattern that does not demand a vertex refinement.

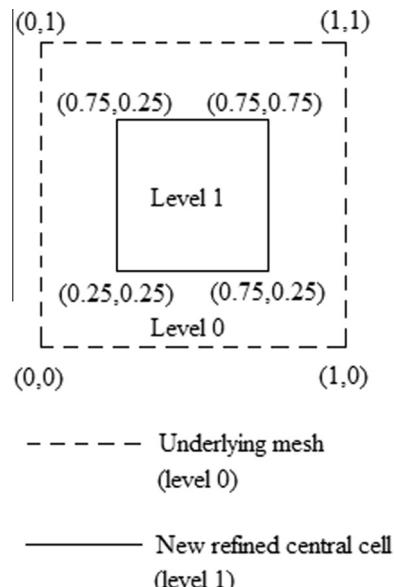


Fig. 4. Local mesh refinement example.

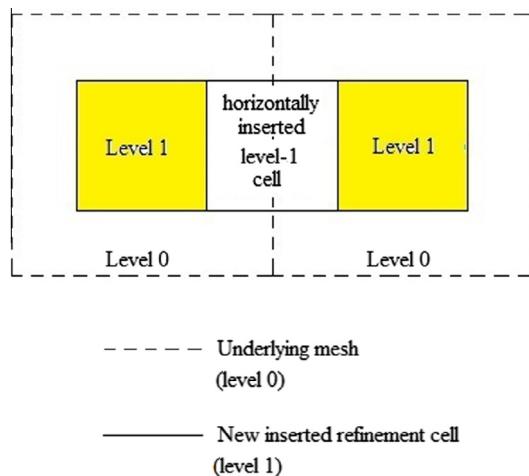
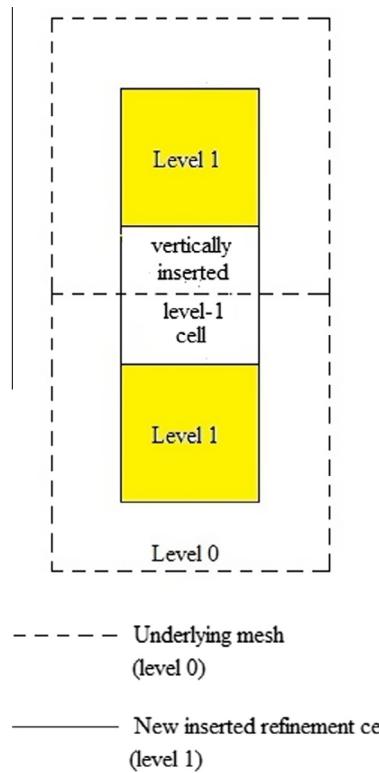


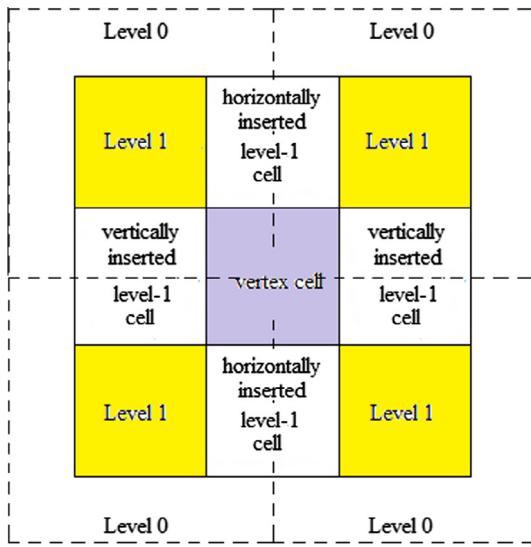
Fig. 5. Application of horizontal inserting refinement rule.

The inserting and vertex refinement rules are both of auxiliary nature, which do not create new nodes but generate new level-1 cells. When level-1 nodes and cells are generated as a result of the sequential applications of the central, inserting, and vertex refinement rules, the cell-wise and node-wise data structures, which are described in Section 3.1, should be both updated using the local cell and node numbering conventions, which are also elaborated in Section 3.1 for the underlying mesh. In the case of Fig. 8 where some level-1 nodes and cells are missing due to the L-shaped refinement pattern, an assigned value of 0 has to be used for data structure updating purpose, which indicates their non-existence.

After the adaptive level-1 mesh is locally generated and the data structure is also accordingly updated, the monitoring parameter for some refined cells is likely insufficiently decreased yet, demanding further local mesh refinement. As demonstrated in Fig. 9, on the basis of level-1 central refinement, another four new nodes are generated, leading to an even higher grid resolution, which is referred to as “level-2” cell. This level-2 cell has one quarter of the area of a level-1 cell, remains in the central of its “parent” and even “grandparent” cell. Escorting the level-2 central refinements, the auxiliary inserting and possibly together with vertex refinement rules may also apply whenever necessary. As update, all the mesh adaptive information should be registered in the AMR cell tree data structure. In the present study, it should be remarked that, in order to reduce the local monitoring parameter value towards its threshold, the mesh adaptation loop undergoing sequentially the



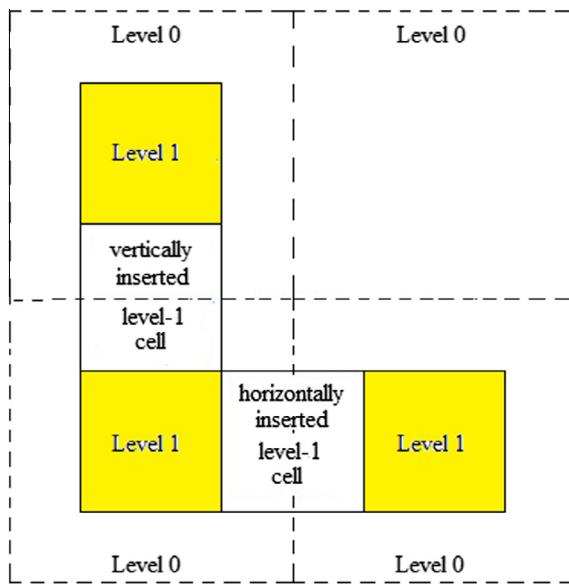
**Fig. 6.** Application of vertical inserting refinement rule.



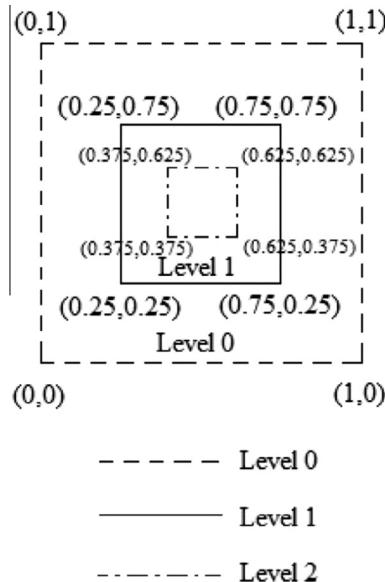
**Fig. 7.** Resulting level-1 refinement pattern with a vertex cell generated.

central, inserting, and vertex refinements can be recursively implemented so as to get the critical region locally intensively refined up to *level-3*.

Each adapted mesh may involve a set of grid resolution levels. The mesh adaptive data storage has to be carefully managed so that the information of a resulting adaptive grid can be recorded in a complete and irredundant fashion and, then, efficiently conveyed to the subsequent LBM computation process.



**Fig. 8.** Resulting level-1 refinement pattern with no vertex cell generated.

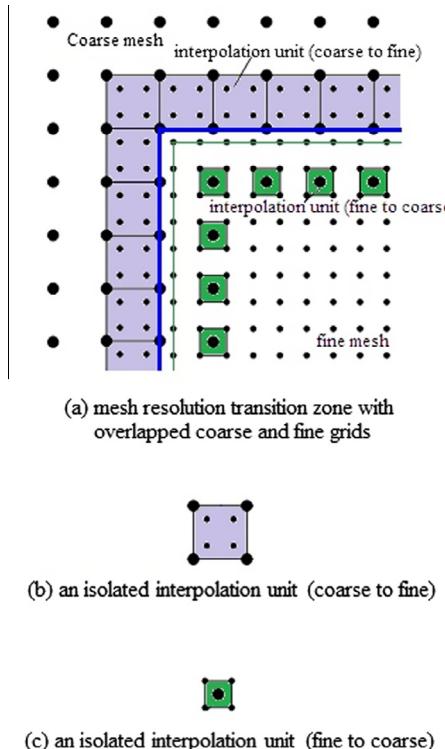


**Fig. 9.** Schematic of three nested mesh levels.

### 3.4. Solution interpolation

#### 3.4.1. Interpolation region

The adaptive mesh generated by the AMR rules in this study generally may include some grid resolution overlay bands, each of which features co-existence of coarse and fine cells in an overlapping manner. Fig. 10(a) illustrates such a coarse and fine mesh overlapping region, in which the bigger points represent the nodes on the coarse mesh while the smaller points stand for the nodes on the fine mesh. It should be remarked that the LBM collision and streaming processes can directly apply only between lattices with an identical grid resolution. In other words, corresponding to the mesh shown in Fig. 10(a) where the “parent” (coarse) and “child” (fine) cells are overlapped at the interface band, two rounds of LBM node-by-node computations are required respectively for the following two groups of nodes:

**Fig. 10.** Schematic of an overlay region.

**Group 1:** the bigger nodes located on the upper and left side of the thick solid line (inclusive), for which the LBM framework uses  $\Delta t$  as time step.

**Group 2:** the smaller nodes located on the lower and right side of the thin solid line (inclusive), for which the LBM framework uses  $\frac{\Delta t}{2}$  as time step.

For the sake of convenience, the following definitions are also useful:

**Group 1\_BDR:** the bigger nodes belonging to Group 1 and exclusively located on the thick solid line.

**Group 1\_SUPL:** the bigger nodes that do not belong to Group 1 and are surrounded by four fine grid nodes (e.g., bigger points on Fig. 10(a) that are located in the center of highlighted smaller cells).

**Group 2\_BDR:** the smaller nodes belonging to Group 2 and exclusively located on the thin solid line.

**Group 2\_SUPL:** the smaller nodes that do not belong to Group 2 and are located inside a coarse cell formed by four coarse grid nodes belonging to Group 1\_BDR (e.g., smaller points on Fig. 10(a) that are located in highlighted bigger cells).

Clearly, when running LBM at the nodes belonging to Group 1\_BDR, the solution at the nodes belonging to Group 1\_SUPL are requested although the latter nodes are not subjected to the LBM computation. Therefore, the solutions at the nodes in Group 1\_SUPL can be obtained only through an interpolation approach. Likewise, LBM can be applied to the nodes in Group 2\_BDR, which demands the solution at some nodes belonging to Group 2\_SUPL; thus, the solution at all nodes belonging to Group 2\_SUPL needs to be determined by interpolation.

As a result, two types of interpolation scenarios are encountered in the present AMR algorithm. One is to seek the solution at four nodes of a fine cell via interpolation based upon the LBM solution at the four vertices of the “parent” (coarse) cell, as illustrated in Fig. 10(b). The other interpolation scenario is, as depicted in Fig. 10(c), to obtain the solution at a coarse node through interpolation, which uses the LBM solution at the four surrounding vertices that belong to a “child” (fine) cell.

### 3.4.2. Interpolation methods

The MRT-LBM scheme exhibits the second-order accuracy. When selecting an interpolation approach for the AMR-LBM framework, both the accuracy and the expense of the interpolation method should be assessed. It is undoubtedly that an interpolation formula of less than second-order accuracy may easily lead the subsequent LBM solution to distorting. On the other hand, a sophisticated interpolation strategy ensuring a higher degree of accuracy may turn out to be an unacceptably expensive computation. To resolve such a dilemma, the quadratic bubble function for the momentum interpolation within a

square cell, which was derived by Geier et al. [37] with the aid of finite difference analysis, manifests its advantage in appropriately balancing interpolation accuracy and the computation expense. Extracted from [37], the bubble interpolation approach within the MRT–LBM context results in the following expression of the two components of the momentum,  $\rho u_x$  and  $\rho u_y$ , within a cell:

$$(\rho u_x)_{\text{within\_cell}} = \pi_x(x, y) = a_0 + a_x x + a_{xx} x^2 + a_{xy} x y + a_y y + a_{yy} y^2, \quad (29)$$

$$(\rho u_y)_{\text{within\_cell}} = \pi_y(x, y) = b_0 + b_x x + b_{xx} x^2 + b_{xy} x y + b_y y + b_{yy} y^2, \quad (30)$$

where  $a_0$ ,  $a_x$ ,  $a_{xx}$ ,  $a_{xy}$ ,  $a_y$ ,  $a_{yy}$ , and  $b_0$ ,  $b_x$ ,  $b_{xx}$ ,  $b_{xy}$ ,  $b_y$ ,  $b_{yy}$  are all uniquely determinable coefficients that depend only upon the coordinates of the four vertices of the cell and the nodal LBM momentum solution at the four vertices. Obviously, Eqs. (29) and (30) do not involve the momentum solution beyond the four vertices of the cell, neither do they request the momentum solution at other time steps, as indicated in [37]. These advantages render great convenience for interpolation of the momentum at any point within the cell, such as the cases depicted in Figs. 10(b) and (c). In terms of accuracy, it was demonstrated in [37] that the interpolated momentum via Eqs. (29) and (30), which are built upon the central finite difference scheme for the discretization of the momentum, can unconditionally achieve the second order of accuracy.

Simpler than the bubble interpolation for the momentum, the density,  $\rho$ , and the distribution function,  $f$ , can be obtained by using a bilinear interpolation scheme, which suffices for ensuring the density and, in turn, the pressure,  $p$ , (due to the proportionality between  $p$  and  $\rho$  within the LBM framework, as indicated in Eq. (13)) to satisfy the Navier–Stokes equation that involves only the gradient of the pressure. Then, the velocity can straightforwardly determined by:

$$u_x = \frac{(\rho u_x)}{\rho}, \quad (31)$$

$$u_y = \frac{(\rho u_y)}{\rho}. \quad (32)$$

Meanwhile, it should be remarked that, before running the bilinear interpolation process, the distribution function value should be transformed using:

$$f_{af} = f_{ac}^{eq} + \frac{\tau_f - 1}{m(\tau_c - 1)} (f_{ac} - f_{ac}^{eq}), \quad (33)$$

in the case of Fig. 10(b), and

$$f_{ac} = f_{af}^{eq} + m \frac{\tau_c - 1}{\tau_f - 1} (f_{af} - f_{af}^{eq}), \quad (34)$$

in the case of Fig. 10(c). In both Eqs. (33) and (34), the parameter  $m$  represents the ratio of the coarse lattice edge to the fine one ( $m = 2$  remains unchanged for the adaptive refinements employed in the present study),  $\tau$  is the relaxation time, and the subscripts  $c$  and  $f$  stand for the coarse and fine grid resolutions, respectively.

#### 4. Numerical experiments and discussion

To verify the validity and accuracy of the present AMR–IB–LBM, a series of incompressible viscous flows past stationary vertically-aligned bi-cylinders and moving single cylinder in both steady and unsteady states are chosen as numerical experiments. These problems have been studied extensively, and some existing results reported elsewhere will serve here as references for comparison use.

In the present simulations, the Reynolds number (Re) is defined as:

$$\text{Re} = \frac{U_\infty D}{v}, \quad (35)$$

where  $U_\infty$  is the free stream velocity in the stationary-obstacle case, or the relative velocity of the free stream to the moving cylinder,  $D$  is the diameter of cylinder, and  $v$  is the kinematic viscosity of the fluid. The drag force can be calculated using:

$$F_D = - \int_{\Omega} f_x d\nu, \quad (36)$$

where  $f_x$  stands for the  $x$ -component of the force density. Then, the drag coefficient is then defined as:

$$C_d = \frac{2F_D}{\rho U_\infty^2 D}. \quad (37)$$

Similarly, the lift coefficient can be defined as:

$$C_l = \frac{2F_L}{\rho U_\infty^2 D}, \quad (38)$$

where  $F_L$  is the lift force obtained by:

$$F_L = - \int_{\Omega} f_y d\nu, \quad (39)$$

in which  $f_y$  stands for the  $y$ -component of the force density. The pressure coefficient,  $C_p$ , is calculated using:

$$C_p = \frac{2(p_B - p_{\infty})}{\rho U_{\infty}^2}, \quad (40)$$

where  $p_B$  and  $p_{\infty}$  are the pressure on the boundary of cylinder and the far-field pressure, respectively. Similar to Eq. (21),  $p_B$  can be obtained by the following interpolation in the flow field:

$$p_B = \sum_{ij} p(\vec{x}, t) D_{ij} (\vec{x}_{ij} - \vec{x}_B^l) \Delta x \Delta y. \quad (41)$$

In all present cases, the nominal fluid density is set at  $\rho = 1.0$ , and the diameter of the cylinder is 5. The LBM computation occupies a  $512 \times 256$  rectangular domain, which is considered as the underlying or level-0 mesh, and three mesh levels, defined as levels 1, 2, and 3, are allowed for the local refinement purpose when needed. The edge of cell unit on each level is  $\Delta x = \Delta y = 1, 0.5, 0.25, 0.125$ , respectively. The equilibrium distribution function is employed as the boundary condition at all four borders of the rectangular domain.

#### 4.1. Accuracy of the present hybrid method

In order to assess the numerical accuracy of the present highly hybrid model, numerical tests corresponding to a flow past a stationary cylinder at  $Re = 40$  are performed using meshes with different finest cell granularities. Since switching on the AMR option is very likely to result in a set of adaptive meshes that arrive at their respective expected grid resolutions but fail to converge to an exactly identical mesh refinement pattern, an identical multi-level-refinement arrangement has to be imposed for all meshes under test so as to maintain the comparability in the study specifically aimed at numerical accuracy examination.

In the present tests, the IB-LBM simulation occupies a  $512 \times 256$  rectangular domain using four nested mesh levels, defined as levels 0, 1, 2, and 3 with their respective cells sized at  $1 \times 1, 0.5 \times 0.5, 0.25 \times 0.25$ , and  $0.125 \times 0.125$ , respectively. In more detail, the dimensions of the four nested zones are  $[0, 512] \times [-128, 128]$ ,  $[39, 295] \times [-64, 64]$ ,  $[59, 187] \times [-32, 32]$ , and  $[63, 127] \times [-16, 16]$ , with coordinates rounded up to the integer part if necessary; the center of the cylinder is locked at  $(80, 0)$ . The trivial way taken in this series of tests for establishing different finest grid resolutions is that the diameter of the cylinder is allowed for moderate adjustment, which changes from 5 as Test A, to 5.5 as Test B, and 6 as Test C at the end, along with the alteration of the fluid viscosity in order to maintain  $Re = 40$ , while the above-specified multi-level mesh remains unchanged. Let the finest grid resolution,  $\eta$ , be represented by the ratio of the fixed finest edge length (0.125 in the present tests) to the diameter of the testing cylinder. As a result, the three tests will have  $\eta_A = 2.5 \times 10^{-2}$ ,  $\eta_B = 2.273 \times 10^{-2}$ , and  $\eta_C = 2.083 \times 10^{-2}$ , respectively.

According to the evaluation method proposed by Roache [46], suppose  $\phi$  is the exact solution; in presence of a numerical solution,  $\tilde{\phi}$ , which is obtained by using a mesh with  $\eta$  as the index of grid resolution, the dependency of exact solution,  $\phi$ , on relevant factors, including the grid resolution,  $\eta$ , may be expressed as follows:

$$\phi = \tilde{\phi} + \alpha \eta^{\beta}, \quad (42)$$

where both  $\alpha$  and  $\beta$  are constants and, in particular,  $\beta$  represents the accuracy order of the numerical method. Therefore, three sample meshes, which are Cases A, B, and C in the present test series, are required to build a well-posed algebraic system for the solutions of three unknowns,  $\phi$ ,  $\alpha$  and  $\beta$ .

Let the drag coefficient,  $C_d$ , serve as the representative solution under investigation. The key data of the three tests are grouped in Table 1, resulting in  $\beta = 1.94$ . This indicates that the present highly hybrid multi-level-refinement IB-LBM approach approximately owns the accuracy of second order.

#### 4.2. Flow over two stationary vertically aligned cylinders

First, a series of flows over two stationary vertically aligned cylinders are simulated to validate the proposed AMR-IB-LBM method. Not only is this type of flow characterized by Reynolds number, but also by the vertical gap depth,  $g$ , (see Fig. 11), as

**Table 1**

Tests for determining the numerical accuracy.

Test case	Cylinder diameter	Finest grid resolution ( $\eta$ )	Numerical drag coefficient ( $\tilde{\phi}$ )
A	5	2.5e-2	1.4935
B	5.5	2.27e-2	1.5190
C	6	2.08e-2	1.5383

this parameter plays an important role in the degree of interaction between the two cylinders. In practice, the non-dimensional gap space with respect to the cylinder diameter,  $D$ , i.e.,  $g^* = g/D$ , is employed to differentiate the cases within this flow category. The influence of  $Re$  and  $g^*$  on such flows has been investigated by Kang [40]. This study mainly looks into the more sensitive range of  $0.2 \leq g^* \leq 1.5$  on anticipation of distinctive variation of the flow patterns with altered  $g^*$  and  $Re$ .

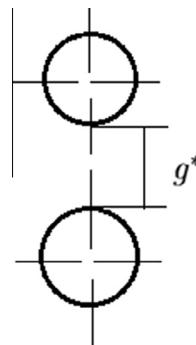
#### 4.2.1. $Re = 40, g^* = 1$

At  $Re = 40$ , the case of two vertically aligned cylinders with a gap of  $g^* = 1$  is examined. Using the AMR-IB-LBM model, Fig. 12 unveils the flow pattern in this bi-cylinder case. It can be observed that the fluid is allowed to flow across the gap area between the two obstructing cylinders and, thus, serves to pushes a bit away the two local recirculation zones, which are respectively generated behind the two cylinders. This pushing action turns each local recirculation zone to a deformed shape, which is now no longer symmetric if individually examined. However, from the global view of the flow pattern, the two local recirculation zones remain symmetric. There are two conjunctive reasons leading to this globally symmetric flow. One reason is that the Reynolds number in this case is as low as 40. The other factor is that the gap is relatively large ( $g^* = 1$ ), which is comparable to  $D$ , the characteristic length for the flow. Owing to this considerable gap size, the flow passes through, playing a role of partitioning the flow domain at the middle horizontal line, and forming two eddies located respectively in the upper and lower parts that have little chance of mutual intervention. In consistency with the global symmetry exhibited in Fig. 12, Fig. 13 shows the adaptively generated mesh that looks overall symmetric as well. The closer to the two obstacles, the finer grid; also, locally intensified mesh refinement can be found in the flow re-circulation zones.

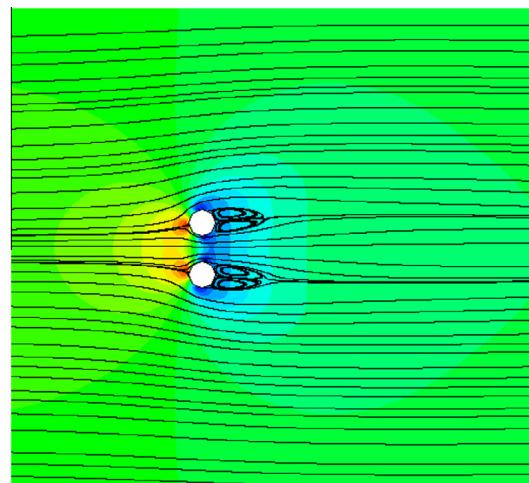
Fig. 14 reveals that the drag and lift coefficients of the two cylinders remain nearly unchanged with time. All the present AMR-IB-LBM results confirm the so called “steady weak flow pattern” for the flow around two vertically aligned cylinders at  $g^* = 1$  and  $Re = 40$ , as elaborated in [40].

#### 4.2.2. $Re = 70, g^* = 0.5$

Following the precedent simulation of the flow over two vertically aligned and gaped cylinders with  $Re = 40$  and  $g^* = 1$ , here the Reynolds number is increased to  $Re = 70$  while the gap is decreased to  $g^* = 0.5$ , so that an unsteady and asymmetric



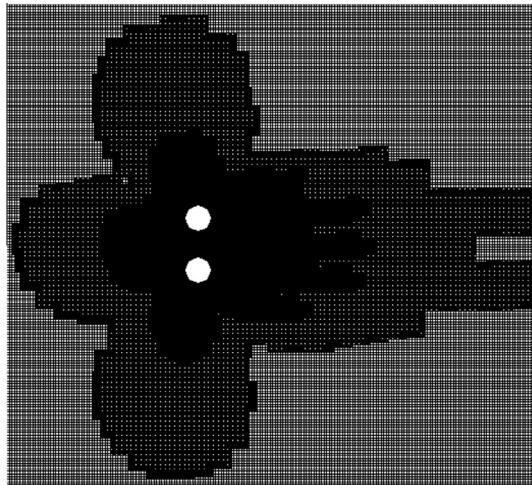
**Fig. 11.** Schematic of two vertically aligned cylinders.



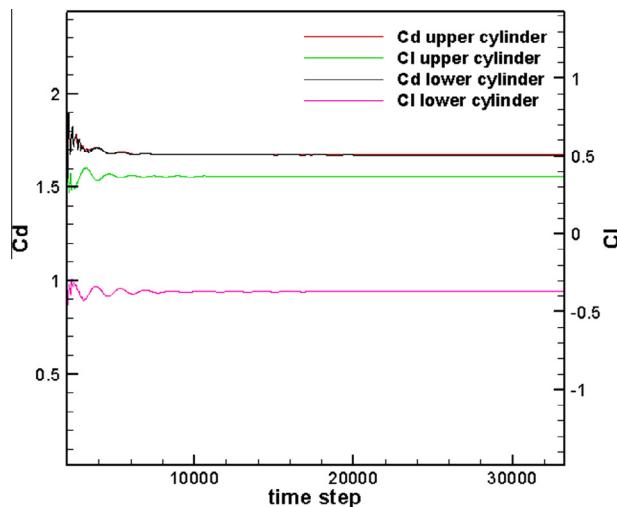
**Fig. 12.** Instantaneous streamline contours at  $Re = 40, g^* = 1$ .

flow pattern can be anticipated. The vorticity contours obtained by the present numerical method no longer look symmetric, as demonstrated in Fig. 15, the adaptive mesh shown in Fig. 16 has no global symmetry, either. Also, the local mesh refinement trace is found practically in accordance with the vortex shedding zones, which indicates the present AMR algorithm can closely reflect important flow characteristics.

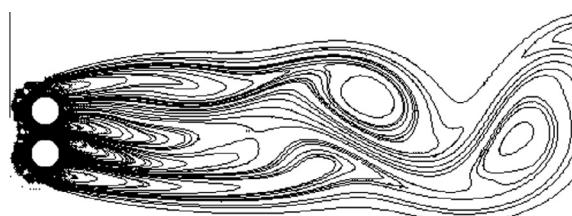
On the other hand, Fig. 17 shows that the time evolution of the drag coefficients for the two cylinders are no longer flat, indicating the flow now becomes unsteady. Moreover, unlike the case of  $Re = 40$  and  $g^* = 1$  where the drags on the upper and



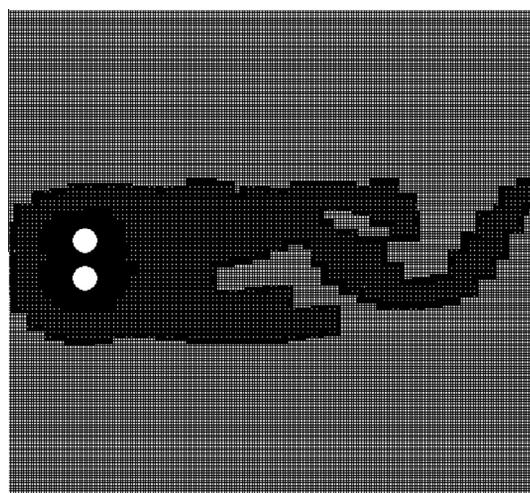
**Fig. 13.** AMR mesh at  $Re = 40$ ,  $g^* = 1$ .



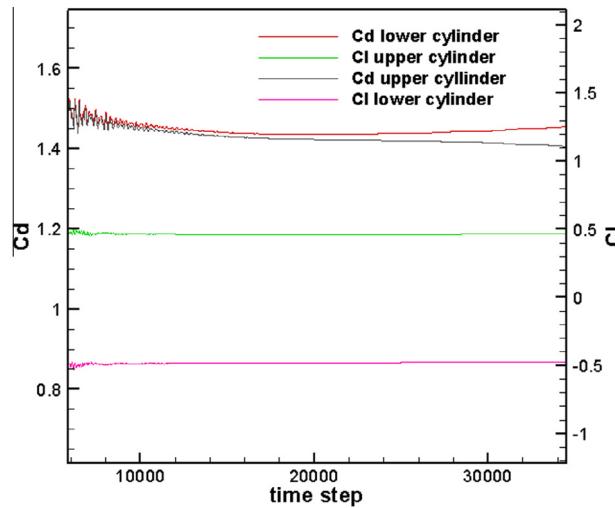
**Fig. 14.** Evolution of drag and lift coefficients for two cylinders at  $Re = 40$ ,  $g^* = 1$ .



**Fig. 15.** Instantaneous vorticity contours at  $Re = 70$ ,  $g^* = 0.5$ .



**Fig. 16.** AMR mesh at  $Re = 70$ ,  $g^* = 0.5$ .



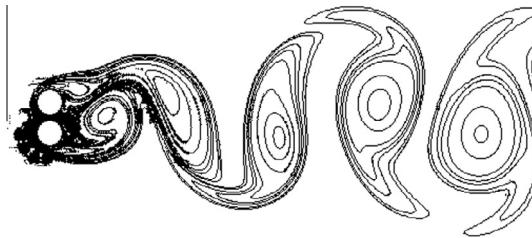
**Fig. 17.** Evolution of drag and lift coefficients for two cylinders at  $Re = 70$ ,  $g^* = 0.5$ .

lower cylinders coincide with each other (see Fig. 14), now the drag coefficient of the lower cylinder is remarkably larger than that of the upper one (see Fig. 17) as iteration time elapses. This suggests that the flow pattern is deflected to the side of higher drag coefficient cylinder, i.e., the lower cylinder, as seen in Fig. 15. The flow deflection phenomenon observed in this simulation case echoes the so called “deflected wake pattern”, as described in [40].

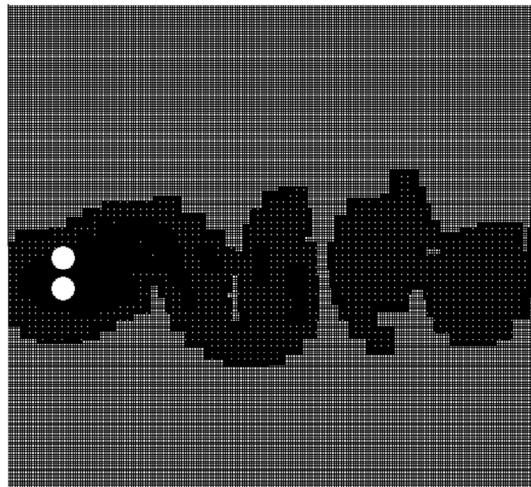
#### 4.2.3. $Re = 100$ , $g^* = 0.2, 0.7, 1.5$

Now, Reynolds number is further increased to  $Re = 100$ , so that the flow is allowed to become more sensitive to the gap depth of the two cylinders. A series of test cases with  $g^* = 0.2, 0.7$  and  $1.5$  were investigated in [40], and the resulting flow patterns are referred to as “single bluff-body pattern”, “flip-flopping pattern”, and “in-phase-synchronized pattern”, respectively. In this AMR-IB-LBM study, these three representative cases are chosen for a further examination on the capability of the present method.

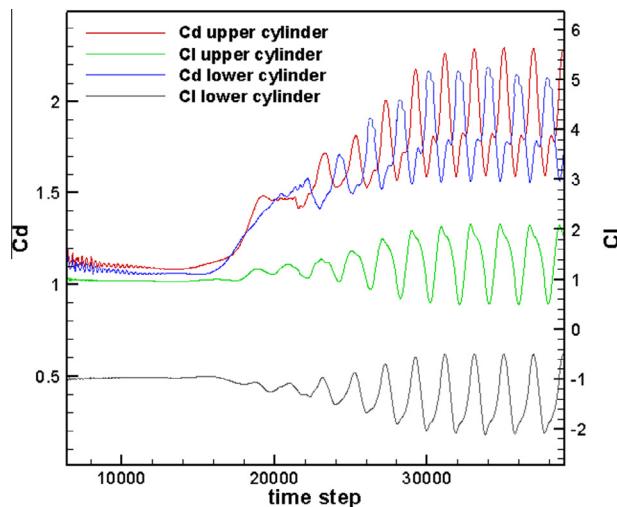
When both cylinders are placed very close with each other, e.g.,  $g^* = 0.2$ , the flow within the gap becomes too weak to affect the flow pattern over the whole domain. The flow behaves as if it is passing over a single bluff-body with a characteristic length of twice as large as the diameter of a single cylinder. A street of shedding vortices arise behind the twin-cylinders, as demonstrated in Fig. 18. Meanwhile, the local mesh refinement pattern shown in Fig. 19 looks in consistency with the vortex-shedding pattern, which reveals the effectiveness of the present grid adaptive technique. A further look at the evolution of drag and lift coefficients, as depicted in Fig. 20, indicates the flow periodicity, which resembles the flow at  $Re = 100$  over a single cylinder, as analyzed in [40]. Thus, the structure of this flow is named “single bluff-body wake pattern”.



**Fig. 18.** Instantaneous vorticity contours at  $\text{Re} = 100$ ,  $g^* = 0.2$ .



**Fig. 19.** AMR mesh at  $\text{Re} = 100$ ,  $g^* = 0.2$ .

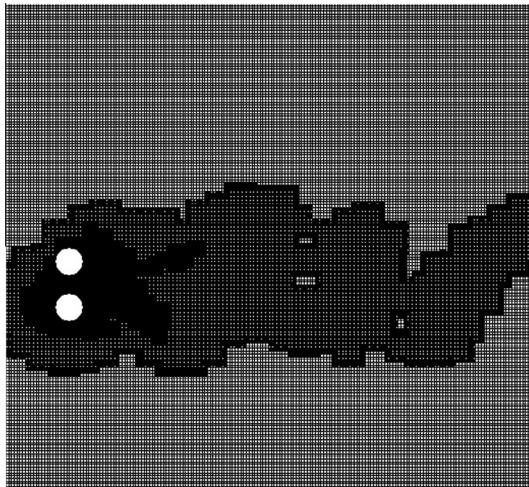


**Fig. 20.** Evolution of drag and lift coefficients for two cylinders at  $\text{Re} = 100$ ,  $g^* = 0.2$ .

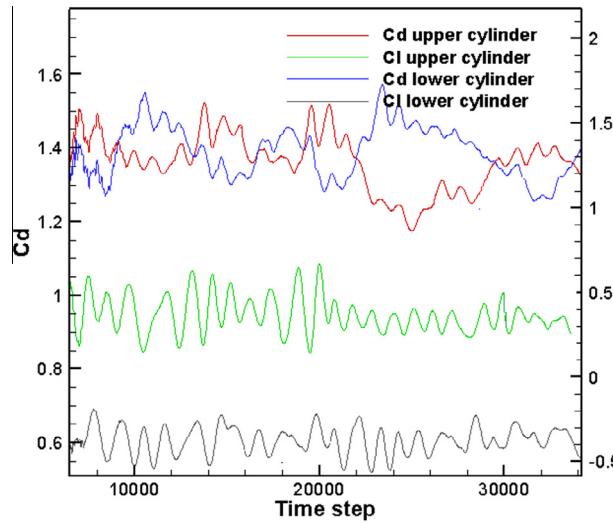
When further increasing the gap spacing to  $g^* = 0.7$ , the flow is no longer periodic but appears unsteady (see Fig. 21). The adaptive mesh shown in Fig. 22 owns no periodic pattern, either. Both drag and lift coefficients undergo irregular changes as observed in Fig. 23. The investigation reported in [40] showed that the flow between the two cylinders deflects to the side of higher-frequency cylinder, and this cylinder sustains a higher drag coefficient than the other. Moreover, both cylinders take irregular turns in experiencing the higher frequency, which indicates that the two wakes randomly flip-flop behind the cylinders, and is referred to as “flip-flopping wake pattern”.



**Fig. 21.** Instantaneous vorticity contours at  $\text{Re} = 100$ ,  $g^* = 0.7$ .



**Fig. 22.** AMR mesh at  $\text{Re} = 100$ ,  $g^* = 0.7$ .

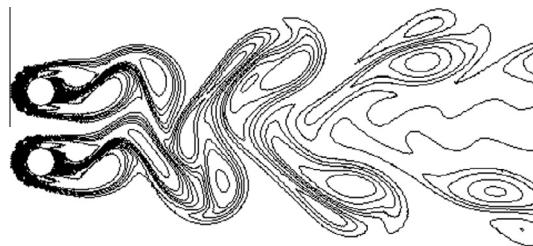


**Fig. 23.** Evolution of drag and lift coefficients for two cylinders at  $\text{Re} = 100$ ,  $g^* = 0.7$ .

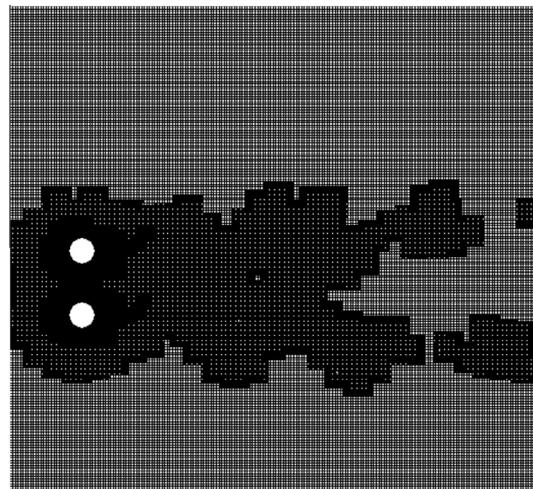
In the case of  $g^* = 1.5$ , the two wakes finally become synchronized (see Figs. 24 and 26). The lift coefficients for both cylinders are in phase, however the drag coefficients are out of phase. Therefore, this flow structure is named “*in-phase-synchronized wake pattern*”. The vortices that looks shedding behind both cylinders are in nearly the same phase with vortices tending to merge on a pair-by-pair basis. The adaptive mesh shown in Fig. 25 confirms the phase synchronization as well.

Additionally, Table 2 performs, for all bi-cylinder test cases in this study, a detailed quantitative comparison against the results reported in [40], and the effectiveness of the present hybrid model is revealed.

Using the AMR-IB-LBM approach, three different flow patterns are all precisely observable, and the details of the present results, including the streamline, vorticity contour, drag and lift coefficients, agree in a satisfactory manner with those



**Fig. 24.** Instantaneous vorticity contours at  $\text{Re} = 100$ ,  $g^* = 1.5$ .

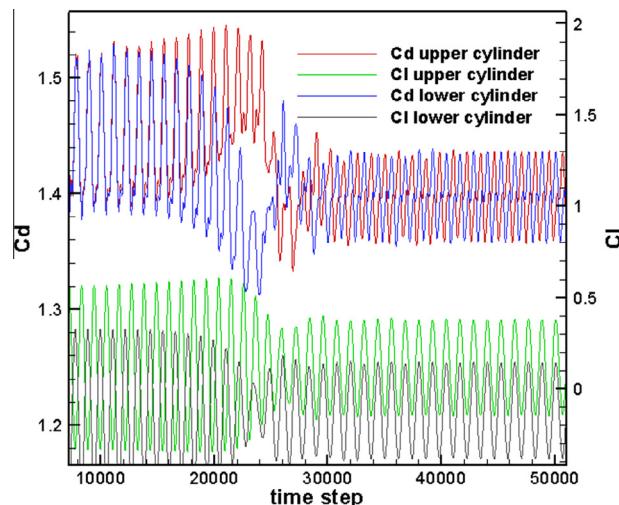


**Fig. 25.** AMR mesh at  $\text{Re} = 100$ ,  $g^* = 1.5$ .

reported by [40], and rational mesh refinement pattern can be achieved by the present AMR algorithm in all the cases tested in here. This justifies the validity and reliability of the AMR-IB-LBM method developed in this study.

#### 4.3. Steady flow over a moving cylinder at $\text{Re} = 40$

The test cases in Section 4.1 have proved that the AMR-IB-LBM method can produce reliable numerical results when simulating complex stationary-obstacle problems. Here, the scenario is modified such that the free stream velocity keeps



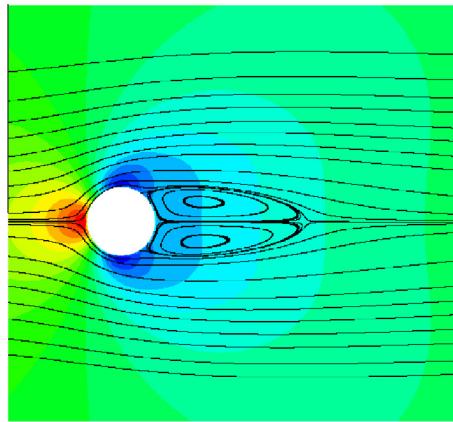
**Fig. 26.** Evolution of drag and lift coefficients for two cylinders at  $\text{Re} = 100$ ,  $g^* = 1.5$ .

**Table 2**

Drag and lift coefficient comparison at different Re and gap space.

	Upper cylinder			Lower cylinder	
	$C_d$	$C_l$	$C_d$	$C_l$	
Re = 40, $g^* = 1$	Kang [40]	1.70	0.38	1.70	-0.38
	Present	1.69	0.38	1.69	-0.38
Re = 70, $g^* = 0.5$	Kang [40]	1.37	0.46	1.49	-0.46
	Present	1.40	0.47	1.46	0.47
Re = 100, $g^* = 0.2$	Kang [40]	[1.63, 2.19]	[0.41, 1.59]	[1.63, 2.19]	[-1.51, -0.39]
	Present	[1.59, 2.28]	[0.52, 2.04]	[1.56, 2.19]	[-2.05, -0.52]
Re = 100, $g^* = 0.7$	Kang [40]	[1.20, 1.54]	[0.08, 0.61]	[1.21, 1.59]	[-0.68, -0.04]
	Present	[1.18, 1.52]	[0.13, 0.67]	[1.25, 1.57]	[-0.58, -0.19]
Re = 100, $g^* = 1.5$	Kang [40]	[1.40, 1.46]	[-0.10, 0.46]	[1.40, 1.46]	[-0.46, 0.10]
	Present	[1.36, 1.44]	[-0.12, 0.39]	[1.36, 1.44]	[-0.39, 0.12]

Note: this table shows the drag and lift coefficients with only two decimal digits, as limitation exists for extracting the data from the time evolution figures in Kang [40].



**Fig. 27.** AMR results of streamlines and pressure distribution.

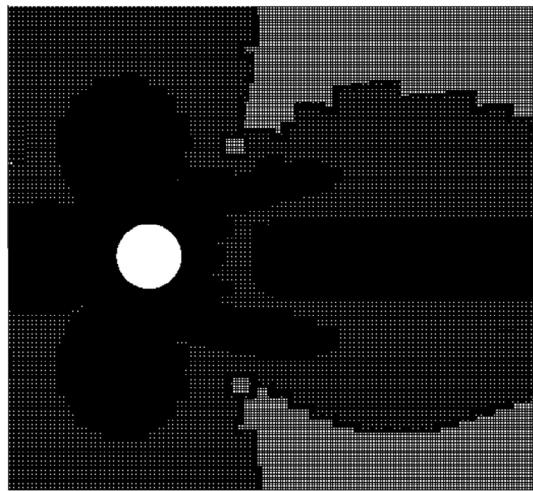
$U_\infty = 0.095$  while a cylinder also moves, at a constant speed of  $U_{object} = -0.005$ , towards the incoming flow. This modification yields a new type of scenario for verifying the effectiveness of the present AMR-IB-LBM model in simulating a flow associated with an immersed moving obstacle. It is well-known that the flow over a stationary cylinder at  $Re = 40$  features symmetry about the horizontal line passing through the center of cylinder. Here, the modification makes the incoming flow velocity relative to the moving cylinder remain 0.1, still resulting in  $Re = 40$ . Thus, the standard flow pattern for  $Re = 40$  is anticipated in this moving cylinder case.

The pressure distribution (as background) and streamline pattern (in black lines) are both shown in Fig. 27. A pair of symmetric recirculating eddies behind the cylinder is demonstrated, while there is no indication of flow penetration across the boundary of the immersed object, confirming the success of incorporation of IB in LBM. Also, Fig. 28 shows the adaptively refined mesh. The finest grid resolution appear in the proximity of the cylinder as well as in the recirculation zone. Both figures feature overall symmetry, confirming the flow remains symmetric at  $Re = 40$  despite the motion of the cylinder.

The drag coefficient and length of recirculation zone are grouped in Table 3 with comparison between the AMR-IB-LBM results and other available references, which shows that the present results are within the range of values reported by the references. At the end, Fig. 29 demonstrates the comparison plot of the pressure coefficient results obtained by the present model and the existing reference, and their agreement appears remarkable.

#### 4.4. Unsteady flow over a moving cylinder at $Re = 100$

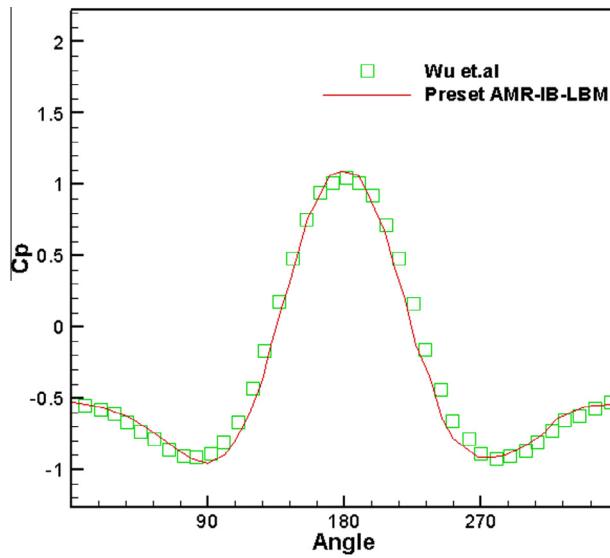
Employing all moving-cylinder conditions used in the precedent case except that the Reynolds number is now increased to  $Re = 100$ , which would lead to the well-known Karman vortex street, the effectiveness of AMR-IB-LBM will be verified in the case of unsteady flow.



**Fig. 28.** AMR mesh at  $Re = 40$ .

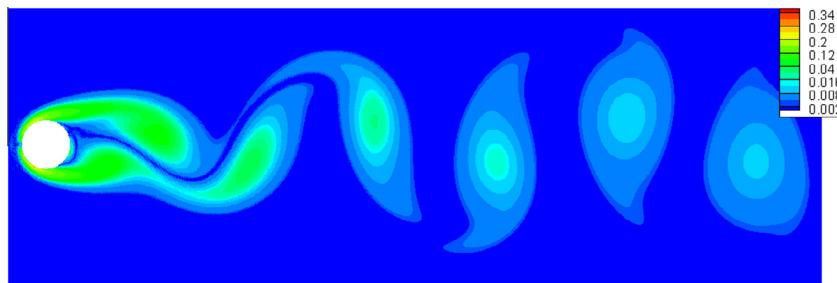
**Table 3**  
Comparison of drag coefficients and length of recirculation zone at  $Re = 40$ .

References	Drag coefficient $C_d$	Length of detected recirculation zone $L$
Fornberg [41]	1.498	2.25
Shukla et al. [42]	1.550	2.34
Niu et al. [35]	1.589	2.26
Wu and Shu [36]	1.554	2.30
AMR	1.561	2.28

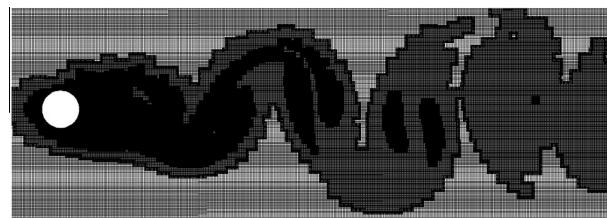


**Fig. 29.** Comparison of pressure coefficient.

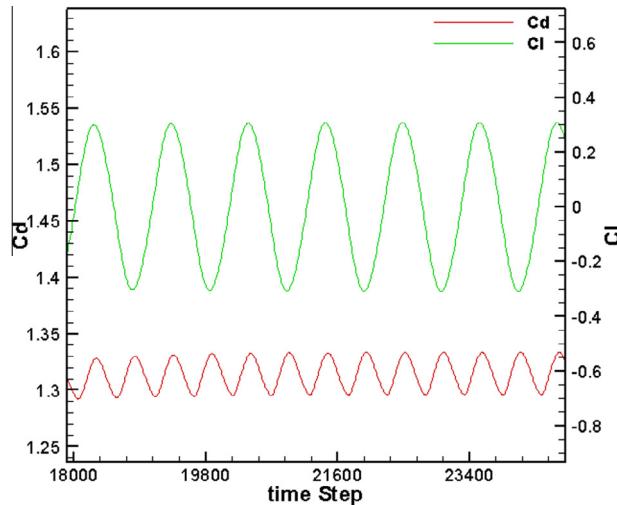
Figs. 30 and 31 show the vorticity contours and the corresponding final adaptive mesh, respectively. In the area behind the moving cylinder, the closer to the obstacle, the larger vorticity, and the more intensified local mesh refinements. As predicted, the adaptive mesh in Fig. 31 shows the trace of Karman vortex street, and this vortex street can be clearly observed in Fig. 30. The vortices are shedding at a constant frequency, which can be proven through the evolution of the lift and drag coefficients, as depicted in Fig. 32. The oscillation of the two coefficients in Fig. 32 looks periodic, which demonstrates the flow field varies periodically when the vortices are shed from the cylinder.



**Fig. 30.** Vorticity contours on final adaptive mesh for  $Re = 100$ .



**Fig. 31.** AMR mesh at  $Re = 100$ .



**Fig. 32.** Evolution of drag and lift coefficients at  $Re = 100$ .

Furthermore, the Strouhal number  $St = \frac{f_g D}{U_\infty}$  is employed to study this oscillating flow, as shown in [Table 4](#). A detailed comparison of the present drag coefficient and Strouhal number results against other available data demonstrates a satisfactory degree of agreement. This suggests that AMR-IB-LBM can achieve accurate simulation results for unsteady flows.

#### 4.5. Simulation cost and accuracy comparison

To further demonstrate the advantage of AMR-IB-LBM, flows over two vertically aligned cylinders at  $Re = 40$  and a moving cylinder at  $Re = 100$  are considered as two benchmark cases representing the stationary- and moving-obstacle cases, and some statistics of these two tests are grouped in [Tables 5 and 6](#), respectively. Note that this study used the Intel Core i5-3337U CPU for all test cases and, in both tables, the drag coefficient is selected as index of accuracy while the numbers of nodes, cells, and computation time units are counted as representation of computation expenses.

Unfolding from an identical coarse mesh with  $\Delta x = \Delta y = 1$  for both cases, two different meshing techniques are compared, namely, adaptive mesh refinement (AMR), and uniformly refined mesh. In a relative sense, the results obtained by using the

**Table 4**Comparison of drag coefficients and Strouhal number for  $Re = 100$ .

References	Drag coefficient $C_d$	Strouhal number $St$
Wu and Shu [36]	1.364	0.163
Saiki and Biringen [43]	1.26	0.171
Sui et al. [44]	1.438	0.166
Russell and Wang [45]	1.43	0.175
AMR	1.310	0.172

**Table 5**Case of two vertically aligned cylinders at  $Re = 40$ .

	Uniform coarse mesh	AMR	Uniformly refined mesh
Drag coefficient of the upper cylinder	1.772	1.708	1.702
Number of nodes	131,072	438,837	8,388,608
Number of cells	130,505	430,961	8,339,520
Number of computation time units	265	3514	15,237

**Table 6**Case of a moving cylinder at  $Re = 100$ .

	Uniform coarse mesh	AMR	Uniformly refined mesh
Drag coefficient	1.455	1.310	1.280
Number of nodes	131,072	470,077	8,388,608
Number of cells	130,505	462,694	8,339,520
Number of computation time units	261	3712	14,681

uniformly refined mesh may be assumed with the best accuracy, as it has the finest grid covering the entire computational domain, preventing numerical interpolation error from taking place due to the transition of grid resolution. However, uniform refinement can also be viewed as aimless refinement, which leads to fairly significant increase in mesh size and, in turn, exponential increase in computational cost.

For example, in the steady flow case of  $Re = 40$  (see Table 5), the uniformly refined mesh requires 15,237 time units for the computation, which is about 4 times of the AMR computation time. This is because the AMR mesh size is only about  $\frac{1}{20}$  of the uniformly refined mesh, which indicates AMR can greatly save computational resources. In terms of solution accuracy, the drag coefficients resulting from the AMR and uniform refinement are respectively 96.39% and 96.04% of that corresponding to the uniform coarse mesh, thus, both results obtained with the aid of refinement techniques are comparable and can be considered as reliable. However, if compared with each other, AMR is evidently more recommendable than uniform refinement in the case of steady flow simulation.

Likewise, the comparison shown in Table 6 for the unsteady flow over a moving obstacle at  $Re = 100$  delivers practically the same conclusion as Table 5 has yielded. Achieving nearly the same degree of IB-LBM solution accuracy, AMR outperforms the uniformly mesh refinement option in terms of mesh size and computation time reductions, remaining in the advantageous position in the case of unsteady flow simulation.

## 5. Conclusion and extension

As a novel coupling strategy, this study has developed a hybrid numerical framework that incorporates both the feed-back-forcing-based immersed boundary (IB) model and the bubble-interpolation-function-aided adaptive mesh refinement (AMR) algorithm into the lattice Boltzmann method (LBM). The present hybrid approach is capable of simulating a broad class of incompressible fluid flow problems, including the challenging ones associated with moving obstacles immersed in a flowing fluid. To justify the present AMR-IB-LBM approach, a series of test cases are numerically investigated, including flows over a moving cylinder at  $Re = 40$  and 100 and two stationary gapped cylinders.

Numerical results obtained in this study are discussed in detail and compared with other available data reported in literature. The satisfactory agreement between the present results and those obtained elsewhere, as well as the demonstrated improvement of numerical solution accuracy and reduction of computation time owing to the AMR techniques embedded in the present model makes this AMR-IB-LBM approach look promising for its further applications to engineering problems of more practical interests.

The present hybrid approach has been applied to two-dimensional flow simulations in this study. Since it is sometimes difficult to reduce engineering problems from the real three-dimensional nature to a two-dimensional stylization, enhancing the current AMR-IB-LBM approach with three dimensionality will certainly render its capability significantly enhanced.

Application of this hybrid numerical method to simulations of complex flow problems associated with sophisticated geometries inevitably leads to a large-size computation. This calls for high performance computation technology, such as MPI or GPU, in order to get the AMR-IB-LBM computation of real-world flow problems effectively accelerated.

## Acknowledgments

This study was funded mainly by the Discovery Grant of the Natural Sciences and Engineering Research Council (NSERC) of Canada; the financial support by the Aeronautical Science Fund of China (Grant No. 20111453012) is also acknowledged.

## Reference

- [1] M.J. Berger, J. Oliger, Adaptive mesh refinement for hyperbolic partial differential equations, *J. Comput. Phys.* 53 (3) (1984) 484–512.
- [2] M.J. Berger, P. Colella, Local adaptive mesh refinement for shock hydrodynamics, *J. Comput. Phys.* 82 (1) (1989) 64–84.
- [3] K.G. Powell, P.L. Roe, T.J. Linde, T.I. Gombosi, D.L. De Zeeuw, A solution-adaptive upwind scheme for ideal magnetohydrodynamics, *J. Comput. Phys.* 154 (2) (1999) 284–309.
- [4] C. Shen, J.M. Qiu, A. Christlieb, Adaptive mesh refinement based on high order finite difference WENO scheme for multi-scale simulations, *J. Comput. Phys.* 230 (10) (2011) 3780–3802.
- [5] C.W. Lan, C.C. Liu, C.M. Hsu, An adaptive finite volume method for incompressible heat flow problems in solidification, *J. Comput. Phys.* 178 (2) (2002) 464–497.
- [6] P. McCrorquodale, P. Colella, A high-order finite-volume method for conservation laws on locally refined grids, *Commun. Appl. Math. Comput. Sci.* 6 (1) (2011) 1–25.
- [7] R. Löhner, An adaptive finite element scheme for transient problems in CFD, *Comput. Methods Appl. Mech. Eng.* 61 (3) (1987) 323–338.
- [8] J. Cao, Application of a posteriori error estimation to finite element simulation of incompressible Navier–Stokes flow, *Comput. Fluids* 34 (2005) 972–990.
- [9] J. Wackers, G. Deng, A. Leroyer, P. Queutey, M. Visonneau, Adaptive grid refinement for hydrodynamic flows, *Comput. Fluids* 55 (2012) 85–100.
- [10] J. Wackers, B. Koren, A simple and efficient space-time adaptive grid technique for unsteady compressible flows, AIAA Paper, 2003–3825, 2003.
- [11] H. Ji, F.-S. Lien, E. Yee, A new adaptive mesh refinement data structure with an application to detonation, *J. Comput. Phys.* 229 (30) (2010) 8981–8993.
- [12] K. Miller, R.N. Miller, Moving finite elements. I, *SIAM J. Numer. Anal.* 18 (1981) 1019–1032.
- [13] R. Li, T. Tang, P. Zhang, A moving mesh finite element algorithm for singular problems in two and three space dimensions, *J. Comput. Phys.* 177 (2) (2002) 365–393.
- [14] E.A. Dorfl, L. O'C. Drury, simple adaptive grids for 1-D initial value problems, *J. Comput. Phys.* 69 (1987) 175–195.
- [15] W. Huang, Y. Ren, R.D. Russell, Moving mesh partial differential equations (MMPDES) based on the equidistribution principle, *SIAM J. Numer. Anal.* 31 (3) (1994) 709–730.
- [16] S. Chen, G.D. Doolen, Lattice Boltzmann method for fluid flows, *Annu. Rev. Fluid Mech.* 30 (1998) 329–364.
- [17] D. d'Huméries, Generalized lattice Boltzmann equations, *Rarefied Gas Dyn. – Theory Simul.* 159 (1994) 450–458.
- [18] Z. Guo, T.S. Zhao, Lattice Boltzmann model for incompressible flows through porous media, *Phys. Rev. E* 66 (2002) 036304.
- [19] H.D. Chen, S. Kandasamy, S. Orszag, Extended Boltzmann kinetic equation for turbulent flows, *Science* 301 (2003) 633–636.
- [20] X.X. Guo, C.W. Zhong, C.S. Zhuo, J. Cao, Multiple-relaxation-time lattice Boltzmann method for study of two-lid-driven cavity flow solution multiplicity, *Theor. Comput. Fluid Dyn.* 28 (2) (2014) 215–231.
- [21] G.W. Peng, H.W. Xi, C. Duncan, S.H. Chou, Finite volume scheme for the lattice Boltzmann method on unstructured meshes, *Phys. Rev. E* 59 (4) (1999) 4675–4682.
- [22] D.Z. Yu, R.W. Mei, W. Shyy, A multi-block lattice Boltzmann method for viscous fluid flows, *Int. J. Numer. Methods Fluids* 39 (2002) 99–120.
- [23] X.M. Sun, F. He, A high-order meshless Bhatnagar–Gross–Krooss scheme based on point collocation, *Chin. Phys. Lett.* 21 (2004) 233–236.
- [24] B. Crouse, E. Rank, M. Krafczyk, J. Tolke, A LB-based approach for adaptive flow simulations, *Int. J. Mod. Phys. B* 17 (2002) 109–112.
- [25] P. Neumann, T. Neckel, A dynamic mesh refinement technique for Lattice Boltzmann simulations on octree-like grids, *Comput. Mech.* 51 (2013) 237–253.
- [26] J. Tolke, S. Freudiger, M. Krafczyk, An adaptive scheme using hierarchical grids for lattice Boltzmann multi-phase flow simulations, *Comput. Fluids* 35 (2006) 820–830.
- [27] Z. Yu, L.S. Fan, An interaction potential based lattice Boltzmann method with adaptive mesh refinement (AMR) for two-phase flow simulation, *J. Comput. Phys.* 228 (2009) 6456–6478.
- [28] P. MacNeice, K.M. Olson, C. Moberly, R. de Fainchtein, C. Packer, PARAMESH: a parallel adaptive mesh refinement community toolkit, *Comput. Phys. Commun.* 126 (3) (2000) 330–354.
- [29] J. Wu, C. Shu, A solution-adaptive lattice Boltzmann method for two-dimensional incompressible viscous flows, *J. Comput. Phys.* 230 (2011) 2246–2269.
- [30] H. Ding, C. Shu, A stencil adaptive algorithm for finite difference solution of incompressible viscous flows, *J. Comput. Phys.* 214 (2006) 397–420.
- [31] S.M. Guzik, T.H. Weisgraber, P. Colella, B.J. Alder, Interpolation methods and the accuracy of lattice-Boltzmann mesh refinement, *J. Comput. Phys.* 259 (2014) 461–487.
- [32] C.S. Peskin, The immersed boundary method, *Acta Numer.* 11 (2002) 479–517.
- [33] Z.G. Feng, E.E. Michaelides, The immersed boundary–lattice Boltzmann method for solving fluid–particles interaction problems, *J. Comput. Phys.* 195 (2004) 602–628.
- [34] Z.G. Feng, E.E. Michaelides, Proteus: a direct forcing method in the simulations of particulate flows, *J. Comput. Phys.* 202 (2005) 20–51.
- [35] X.D. Niu, C. Shu, Y.T. Chew, Y. Peng, A momentum exchanged-based immersed boundary–lattice Boltzmann method for simulation incompressible viscous flows, *Phys. Lett. A* 354 (2006) 173–182.
- [36] J. Wu, C. Shu, Implicit velocity correction-based immersed boundary–lattice Boltzmann method and its applications, *J. Comput. Phys.* 228 (2009) 1963–1979.
- [37] M. Geier, A. Greiner, J.G. Korvink, Bubble functions for the lattice Boltzmann method and their application to grid refinement, *Eur. Phys. J.* 171 (2009) 173–179.
- [38] W.X. Huang, H.J. Sung, Simulation of flexible filaments in a uniform flow by the immersed boundary method, *J. Comput. Phys.* 226 (2) (2007) 2206–2228.
- [39] Z.L. Guo, C.G. Zheng, B.C. Shi, Discrete lattice effects on the forcing term in the lattice Boltzmann method, *Phys. Rev. E* 65 (4) (2002) 046308.
- [40] S. Kang, Characteristics of flow over two circular cylinders in a side-by-side arrangement at low Reynolds numbers, *Phys. Fluids* 15 (2003) 2486–2498.
- [41] B. Fornberg, A numerical study of steady viscous flow past a circular cylinder, *J. Fluid Mech.* 98 (1980) 819–855.
- [42] R.K. Shukla, M. Tatineni, X. Zhong, Very high-order compact finite difference schemes on non-uniform grids for incompressible Navier–Stokes equations, *J. Comput. Phys.* 224 (2007) 1064–1094.
- [43] E.M. Saiki, S. Biringen, Numerical simulation of a cylinder in uniform flow: application of a virtual boundary method, *J. Comput. Phys.* 123 (1996) 450–465.

- [44] Y. Sui, Y.T. Chew, P. Roy, H.T. Low, A hybrid immersed-boundary and multi-block lattice Boltzmann method for simulating fluid and moving-boundaries interactions, *Int. J. Numer. Methods Fluids* 53 (2007) 1727–1754.
- [45] D. Russell, Z.J. Wang, A Cartesian grid method for modeling multiple moving objects in 2D incompressible viscous flow, *J. Comput. Phys.* 191 (2003) 177–205.
- [46] P.J. Roache, Perspective: a method for uniform reporting of grid refinement studies, *J. Fluids Eng.* 116 (1994) 405–413.